AFRL-IF-RS-TR-2007-141
Final Technical Report
May 2007

SOFTWARE AND SYSTEMS TEST TRACK ARCHITECTURE AND CONCEPT DEFINITION

The Boeing Company

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

AFRL-IF-RS-TR-2007-141 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

WILLIAM E. MCKEEVER
Work Unit Manager

/s/

JAMES A COLLINS, Deputy Chief
Advanced Computing Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* MAY 2007	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* Jul 06 – Jan 07

4. TITLE AND SUBTITLE

SOFTWARE AND SYSTEMS TEST TRACK ARCHITECTURE AND CONCEPT DEFINITION

5a. CONTRACT NUMBER FA8750-06-C-0213

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER 63781D

6. AUTHOR(S)

James Paunicka, Doug Stuart, Patrick Stokes, Don Wilson, Andrew Vandivort, Ted Bapty, Janos Sztipanovits, Shankar Sastry, Jonathan Sprinkle and Larry Rohrbough

5d. PROJECT NUMBER SSTT

5e. TASK NUMBER BO

5f. WORK UNIT NUMBER 06

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Phantom Works
The Boeing Company
Po Box 516
St Louis MO 63166-0516

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFTC
525 Brooks Rd
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2007-141

12. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 07-241

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This Final Technical Report for the Software and Systems Test Track (SWTT) Architecture and Concept Definition Phase I effort provides documentation on the work accomplished by a Boeing-led team in developing a user Concept of Operations (CONOPS) and Architecture for the emerging SWTT. The objective of the SWTT is to provide an open collaborative research and development environment to demonstrate, evaluate and document the ability of novel tools, methods, techniques and run-time technologies to yield affordable and more predictable software intensive systems.

15. SUBJECT TERMS

Software Testbed, Tool Chains, System Architecture, Software Intensive Systems

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			William E. McKeever, Jr.
U	U	U	UL	50	**19b. TELEPHONE NUMBER** (***Include area code)***

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Table of Contents

List of Figures

List of Tables

1.0 Introduction

1.1. Document Identification

This Final Technical Report for the Software and Systems Test Track (SWTT) Architecture and Concept Definition Phase I effort provides documentation on the work accomplished by a Boeing-led team in developing a user Concept of Operations (CONOPS) and Architecture for the emerging SWTT. This work is part of Air Force Research Laboratory (AFRL) Contract Number FA8750-06-C-0213, in response to Broad Agency Announcement (BAA) 06-13-IFKA. SWTT development is part of the Office of the Secretary of Defense (OSD) Software Intensive Systems Producibility Initiative (SiSPI). This is the final release of the document, and is being delivered as CLIN 0002, CDRL Data Item No. A006 to the AFRL customer.

This Final Technical Report concentrates on the *activities* and *processes* performed and used in developing the CONOPS and Architecture. The *results* of this development work are documented in the following separate volumes:

- Concept of Operations for the Software and Systems Test Track, AFRL Contract Number FA8750-06-C-0213, CLIN 0002, CDRL Data Item No. A003, which describes SWTT characteristics from a user's point of view

- Architecture Framework for the Software and Systems Test Track, AFRL Contract Number FA8750-06-C-0213, CLIN 0002, CDRL Data Item No. A004, which describes the fundamental organization of the SWTT as embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution.

1.2. Document Overview

The document summarizes the activities and processes used in defining the CONOPS and Architecture for the SWTT in this Phase I study effort. This will include a summary of organizations and significant personnel involved, an historical record of significant program events, and information on processes used and trades that have been considered.

The intended audience for this document is the government customer community for SWTT. This includes our primary AFRL customers and possibly other government stakeholders and government support organizations. This might include the Office of the Secretary of Defense (OSD), the primary motivating organization for SiSPI, as well as other government labs which will likely make use of the SWTT during execution of their SiSPI technology development efforts, such as the Army Research Laboratory (ARL) and the Office of Naval Research (ONR).

The document has been prepared by a Boeing-led team that also includes Raytheon, the Embedded Systems Consortium for Hybrid and Embedded Research (ESCHER) Research Institute, Vanderbilt University, and the University of California Berkeley (UC-Berkeley) as subcontracted team mates.

2.0 Project Goals

The SWTT is being developed as an open collaborative research and development environment to demonstrate, evaluate, and document the ability of novel tools, methods, techniques, and run-time technologies to yield affordable and more predictable production of software intensive systems. SWTT is being funded as part of the OSD SiSPI.

The current Phase I study effort to define the CONOPS and Architecture of the SWTT is being executed under the auspices of AFRL's Information Directorate at the Rome Research Site (RRS). In addition to AFRL, the SWTT will support a wide range of Department of Defense (DoD) entities, including research organizations from the US Army and Navy. As a case in point, ARL recently issued a Broad Agency Announcement (BAA) amendment for research into Software Technologies Targeting Interoperability for Systems of Systems. In this BAA, the SWTT is identified as one place where the prototype software from the BAA research effort could be delivered for testing.

The SWTT will be a system where SiSPI software tools and technology researchers can test their research against relevant challenge problems, and where operators of SWTT can perform independent analysis of SiSPI research. This independent analysis would enable facility operators to support acquisition program offices' analysis of the utility of the tools and technologies. Also, SWTT would provide a place where program offices can bring their unsolved problems either for help in solving those problems, for example, by leveraging existing tools and technologies available in the SWTT, or to provide challenges that drive SiSPI research where no such tools or technologies are available.

3.0 Execution

3.1. Organizational Involvement

Organizational elements and personnel who were part of the Boeing-led team during the execution of SWTT Phase I are shown in Figure 1. The program was led out of the Boeing Phantom Works Network Centric Operations (NCO) Thrust organization's Contract Research and Development (CRAD) group. As with all CRAD group programs, Patrick Stokes served as program manager, responsible for monitoring program progress, cost, and schedule. Dr. James Paunicka was Principal Investigator, leading technical development and interactions with other technical teams involved in the program. Dr. Douglas Stuart led the development of the CONOPS documentation and also led development of the researcher surveys that helped inform our team's CONOPS and Architecture work on the program.

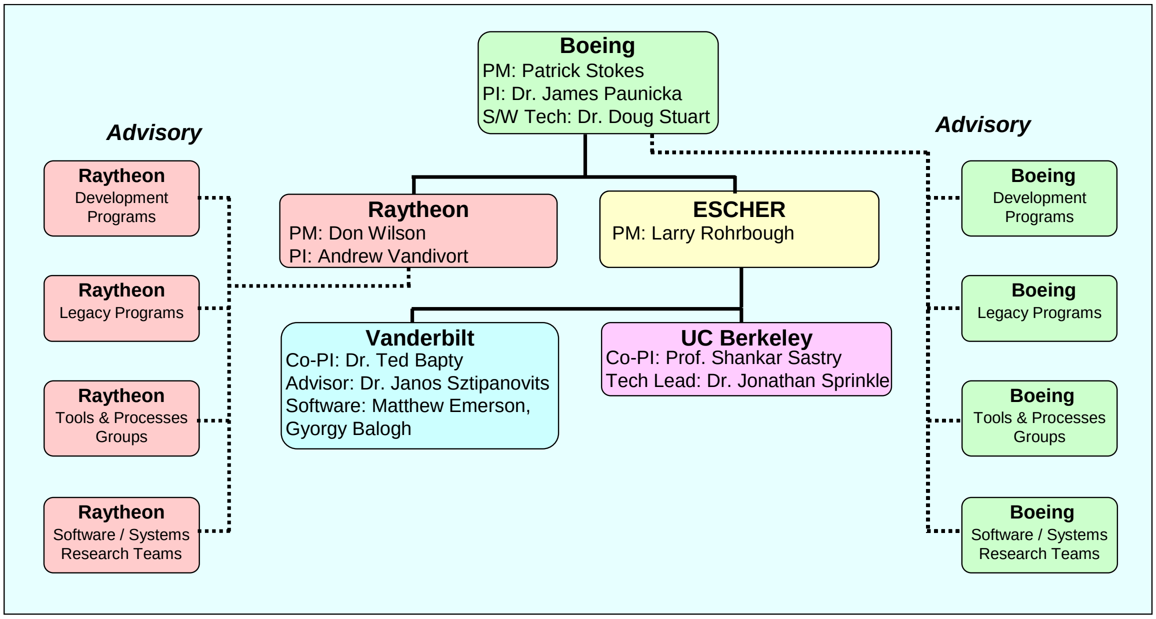

Figure 1: Boeing-Led Team for SWTT Phase I

From Raytheon, Andrew Vandivort contributed greatly to the development of the SWTT CONOPS, and Don Wilson provided strategic technology leadership. From ESCHER, Larry Rohrbough led development of the Architecture concepts worked on the program. From Vanderbilt, Dr. Ted Bapty and Dr. Janos Sztipanovits played key roles in Architecture development, while Matthew Emerson and Gyorgy Balogh worked demonstration concepts synergistic with SWTT goals. At UC-Berkeley, Dr. Jonathan Sprinkle contributed to Architecture development activities with guidance from Dr. Shankar Sastry.

Representatives from various organizations at Boeing and Raytheon, external to our SWTT team, played a part in influencing our CONOPS and Architecture concepts. Software and systems research teams from both companies are a potential source of SWTT infrastructure elements and challenge problems. Tools and processes groups, as well as development and legacy program groups, have provided valuable insight into transitioning research from SiSPI that will be exercised and tested on SWTT.

During execution of the program, our team interacted heavily Mr. William McKeever and Steven Drager of from the AFRL Information Directorate's Advanced Computing

Technology Branch (AFRL/IFTC) of Rome, New York for requirements, schedule, and technical matters. The team also interacted with Mr. Rob Gold from OSD on SiSPI expectations for SWTT. Mr. Glenn Racine from the Army Research Laboratory (ARL) also provided insight into the needs of his emerging SiSPI-funded work, that will likely use test track, including inviting the Boeing team to attend the kickoff for the initial ARL SWTT related research program.

3.2. Historical Summary of Program Events

Program period of performance (PoP) overall was 28 July 2006 to 31 January 2007. During execution, Boeing had weekly coordination telecons with our Raytheon, ESCHER, Vanderbilt, and UC-Berkeley team mates. We also scheduled periodic (approximately bi-weekly) telecons with our AFRL customer (Bill McKeever and Steve Drager). Major events occurring during program execution are shown in Table I.

Table I: Significant Program Events

Event	Date(s)	Location	Organizations / People Involved	Synopsis
PoP Start	28 Jul 2006		Boeing, Raytheon, ESCHER, Vanderbilt, UC-Berkeley	
Consolidated Team Meeting	07 Aug 2006	Boeing Washington DC Office, Arlington, VA	Boeing, Raytheon, ESCHER, Vanderbilt, UC-Berkeley	Final preparations for SWTT Kickoff Meeting
SWTT Kickoff Meeting	08 Aug 2006	ONR Offices in Arlington, VA	AFRL (McKeever, Drager, R. Linderman, Herb Klumpe, et al.), OSD (Gold), ARL (Racine), US Navy, Boeing (Paunicka, Stuart), Raytheon (Wilson, Vandivort), ESCHER (Rohrbough), Vanderbilt (Sztipanovits, Emerson), UC-Berkeley (Sprinkle), Other interested government organizations and researchers	OSD and AFRL briefed vision for the program. AFRL also briefed issues involved in accessing internal AFRL computing systems. Boeing briefed our initial baseline for SWTT CONOPS and Architecture, and our plans for maturation of those concepts in Phase I. Matt Emerson demonstrated concepts based on ESCHER toolchains being used on Boeing Future Combat Systems program activities
Boeing Software Technology Briefing	18 Sep 2006	Boeing St. Louis with Telecon / Webex to	Boeing SWTT Team Members (Paunicka and Stuart, briefers)	Getting SWTT visibility throughout Boeing by being granted the entire agenda for

		Boeing sites throughout the US	Boeing software professionals from across the company	the monthly Boeing Software Technology PAT (Process Action Team) technology session.
Government / Contractor DREN Telecon	16 Aug 2006	Telecon	HPCMC (Eddie Brooks) AFRL (McKeever, Drager) Boeing Team (Boeing, Raytheon, ESCHER, Vanderbilt, UC-Berkeley)	Government DREN domain expert from HPCMC briefed DREN infrastructure and capabilities, access methods, and current user base
JMETC Telecon	17 Oct 2006	Telecon	OSD (Gold) AFRL (Drager, McKeever) Boeing Team (Boeing, Raytheon, ESCHER, Vanderbilt, UC-Berkeley)	Clarified expectations for SWTT and distinguishing it from JMETC
Team Meeting with AFRL	18-19 Oct 2006	Vanderbilt, Nashville, TN (just before Researcher-Focused Workshop)	AFRL (McKeever) on Day 2 Boeing (Paunicka, Stuart) Raytheon (Vandivort) ESCHER (Rohrbough) Vanderbilt (Sztipanovits, Bapty, Balogh) UC-Berkeley (Sprinkle)	Day 1 – Boeing-led team finalized upcoming Workshop briefings; Gyorgy Balogh demo'd C2 Wind Tunnel infrastructure elements for team Day 2 – Boeing-led team briefed current approach to AFRL in preparation for subsequent Researcher-Focused Workshop
Researcher-Focused Workshop	20 Oct 2006	Vanderbilt, Nashville, TN with Telecon / Webex to remote participants across the country	OSD (Rob Gold – telecon) AFRL (McKeever – in person, Drager – telecon) Boeing (Paunicka, Stuart) Raytheon (Vandivort) ESCHER (Rohrbough) Vanderbilt (Sztipanovits, Bapty) UC-Berkeley (Sprinkle) Multiple members of the software research community	Boeing-led team summarized our SWTT approach, our survey motivation, and survey results. This was followed by significant discussion with the research community; researcher feedback aimed mainly on SWTT usability (how to integrate a technology into SWTT for test)
SWTT Mid-Term Review Meeting	02 Nov 2006	AFRL Information Directorate, Rome, NY	AFRL (McKeever, Drager, Bay, R. Linderman, et al.), OSD (Gold), ARL (Racine), Boeing (Paunicka, Stuart), Raytheon (Vandivort), ESCHER (Rohrbough), Vanderbilt (Sztipanovits, Bapty), UC-Berkeley (Sprinkle) Other interested government organizations and researchers	OSD and AFRL briefed vision for the program. Boeing briefed our mid-term baseline for SWTT CONOPS and Architecture.
Infrastructure Demonstration	18 Dec 2006	Vanderbilt, Nashville, TN with Telecon / Webex to Boeing St. Louis	Boeing (Paunicka, Stuart) Vanderbilt (Bapty, Balogh)	Vanderbilt (Balogh) demonstrated matured demo'd C2 Wind Tunnel infrastructure elements for Boeing; some of these

				elements may be leveraged in an initial SWTT instantiation
SWTT Final Review Meeting	19 Jan 2007	ZAI Offices, Rosslyn	AFRL, OSD, US Army, US Navy, Boeing, ESCHER, Vanderbilt, UC-Berkeley, Other interested government organizations and researchers	Boeing-led team briefed final results for CONOPS and Architecture Phase I study
ARL BAA Research Kickoff Meeting	26 Jan 2007	ZAI Offices, Rosslyn	Boeing (Paunicka), Raytheon (Vandivort), UC-Berkeley ARL awardee (Edward Lee), Vanderbilt ARL awardee (Karsai), OSD, ARL, AFRL, ONR	ARL technology awardees briefed their programs, including technologies and execution plan
PoP End	31 Jan 2007		Boeing, Raytheon, ESCHER, Vanderbilt, UC-Berkeley	
BTEC12 (Boeing Technical Excellence Conference 12)	15 Feb 2007	St. Louis, MO	Boeing (Paunicka, Stuart)	Getting SWTT visibility in front of software and systems technologists from across the Boeing company
BSC-2 (Boeing Software Conference 2) Briefing	06 Mar 2007	Long Beach, CA	Boeing (Paunicka, Stuart)	Getting SWTT visibility in front of software technologists from across the Boeing company

3.2.1. Demonstrations

Demonstrations at program-wide meetings have played an important role in illustrating potential SWTT operation and concepts. The demos have used infrastructure elements, borrowed from existing systems, that are similar to what our team envisions will be part of a SWTT system. These off-the-shelf elements are available to affordably build initial versions of SWTT. The demonstrations additionally illustrate (1) the ease of use that is expected for research community customers of the test track, and (2) the richness of the operating environment and metrics collection capabilities

3.2.1.1. Kickoff Meeting Demonstration

In the final phase of our Kickoff Meeting presentation session, after our CONOPS and Architecture briefing, Mr. Matt Emerson of Vanderbilt demonstrated a system that allows for simulation and metrics collection of a system of systems environment. In this environment, there is a capability to plug in application and system components at various levels of completeness, from initial models to executing code. The concepts, infrastructure, and tool elements are partially based on entities available today from ESCHER. The particular capabilities that were demonstrated are based on current toolchains and other infrastructure currently being used on the Boeing Future Combat Systems program activities

3.2.1.2. Final Review Demonstration

During our Final Review Meeting presentation, Mr. Gyorgy Balogh of Vanderbilt University presented a demonstration of the Air Force Office of Scientific Research (AFOSR) C2 Wind Tunnel project (see 3.3.5) to illustrate concepts from our CONOPS and Architecture Framework approach and the availability of artifacts that can be leveraged to create the SWTT, as well as to help show the implementability of our SWTT Architecture Framework concepts, in particular the Open Experiment Integration Platform (OEIP). The demonstration included using a graphical editor to model a system under test, and the experimental environment for its evaluation. Model-based tools were then used to generate multiple variations of an experimental system to explore the impact of parameters on the experiment. Finally, the resulting systems were executed with metrics being collected and displayed for analysis (see Figure 2). In the demonstration, a system of systems scenario was integrated and simulated. The systems integrated included 2 air vehicles, a network, and a human-operated command and control system. At the top of the figure are real-time plots of network activity, fed by data trapped in a metrics collection layer of the composite experimentation system.

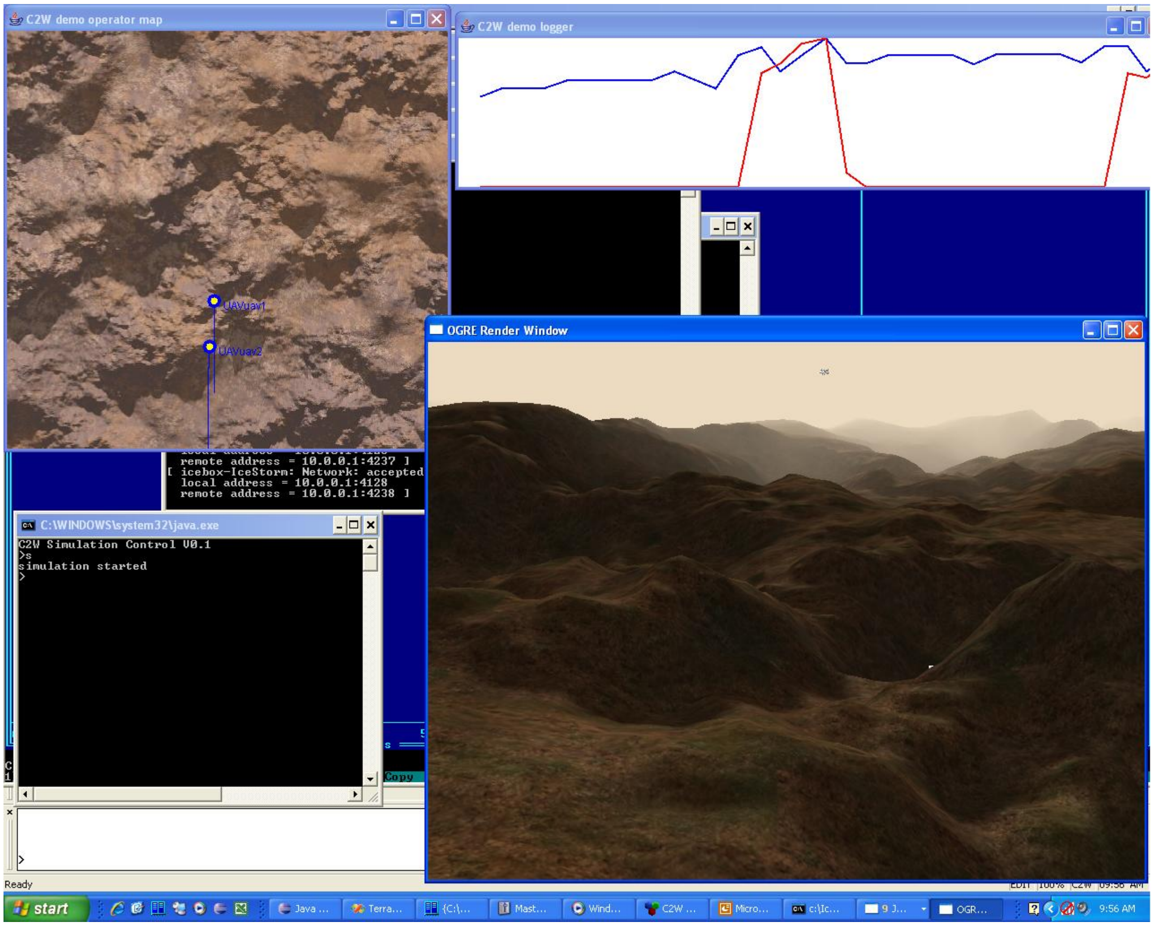

Figure 2: AFOSR C2 Wind Tunnel Composite Experimentation Display

The demonstration showed that by selecting standards based integration layers, something representative of an OEIP can be built using open-source research components. The primary challenge in creating the OEIP is not the physical computation platform (or cluster) where the SWTT would run, but the software infrastructure that can be deployed on a variety of platforms. The architecture of the OEIP for system-of-systems research is inherently heterogeneous. An OEIP needs to include a carefully selected suite of simulation integration, component integration, instrumentation and emulation platforms that are seamlessly integrated for experimentation. However, creating a flexible, high-performance OEIP for the SWTT is a feasible task that does not require huge investment: the challenge is a well designed integration concept.

Also, flexible experiment specification and integration on the top of complex integration platforms benefits greatly from model-based approaches. The demonstration showed an initial example for model-based specification and generation of experiments.

3.3. Processes Used

3.3.1. Team Collaboration

Throughout the execution of the program, team coordination was enabled through the use of the following tools and methods:

- Weekly coordination and planning telecons and Webex's were hosted by Boeing, allowing full participation from our extended (organizationally and geographically) team, including
    - Boeing in St. Louis, Missouri
    - Raytheon in Tucson, Arizona
    - ESCHER in Washington, DC
    - Vanderbilt University in Nashville, Tennessee
    - UC-Berkeley in Berkeley, California
- For focused final coordination before major program meetings, such as the program Kickoff meeting, the program Final Review, and our Researcher-Focused Workshop, members from across our extended team would arrive early (a day or more) for a consolidated team meeting.
- During Phase I CONOPS and Architecture development, proprietary information was communicated with our team mates with two secure, web-based tools hosted at Boeing:
    - Secure email-like communication of content such as computer file enclosures, telecon / Webex announcements and passwords, and telecon

minutes was enabled with the Boeing-hosted MessageCourier system. This tool has the look and feel of web-based email.

- Secure server access to persistent content such as documents and meeting presentations was enabled with a Boeing-hosted SharePoint site. With SharePoint, which has the look and feel of shared server folders, files can be checked out and updated by anyone on our extended team.

During Phase I execution, the various organizations in our extended team had particular responsibilities, summarized in Figure 3.

- **Boeing**
  - **Challenge Problems definition (to ensure sufficiently rich CONOPS and Architecture) with industrial partner Raytheon**
  - **CONOPS definition with industrial partner Raytheon**
  - **Supporting role on architecture development**
  - **Prime contractor responsibilities (leading CDRL development, reporting, program coordination)**
- **Raytheon**
  - **Challenge Problems definition with industrial partner Boeing**
  - **CONOPS definition with industrial partner Boeing**
  - **Supporting role on architecture development**
- **ESCHER (with Vanderbilt and UC Berkeley)**
  - **Lead architecture development**
  - **Consult on CONOPS**

Figure 3: Organizational Roles on Boeing-Led Team

3.3.2. Government and Customer Collaboration

Collaboration with our primary AFRL customers on SWTT and other interested government parties took many forms during Phase I execution, including regularly scheduled and pop-up telecons, web-based interactions, and focused presentations.

3.3.2.1. AFRL Collaboration

During Phase I execution, bi-weekly customer coordination calls were scheduled with our AFRL government customers, Steven Drager and William McKeever. These calls were open to, and attended by, our extended team members from Raytheon, ESCHER, Vanderbilt, and UC-Berkeley. During these calls, our team would communicate status, raise questions and issues, and seek feedback from our customers. AFRL would communicate important program status and plans, make us aware of any programmatic issues, suggest approaches to working with and leveraging government infrastructure, and comment on our current approach to SWTT.

A Basic Support for Cooperative Work (BSCW) Internet-based shared workspace site (https://bscw.sei.cmu.edu/bscw/bscw.cgi) was set up by Steven Drager of AFRL early in the programs. This site was used for exchanging data generated by the multiple participants who attended government-hosted SWTT meetings.

The AFRL Jiffy program management system (https://jiffy.rl.af.mil) was used by Boeing to deliver CDRL documentation, including periodic status reports and other technical reports (CONOPS document, Architecture document, Final Technical Report), over the public internet.

Other program telecons were put together by our AFRL customer on an as-needed basis. For example, a telecon to discuss the Joint Mission Environment Test Capability (JMETC) Program, and how JMETC contrasts with SWTT, occurred on 17 October 2006. Here, our extended Boeing team discussed with OSD and AFRL the SWTT vision in comparison with that of JMETC. Although SWTT and JMETC could be seen as having similar goals and possibly even similar infrastructure components, the outcome of our discussions focused on the reality that SWTT will have more general and less program-specific challenge problems than one would find in JMETC. JMETC will be focused on solving specific acquisition program problems with focused challenge problems applicable to those programs. The focus, though, of SWTT challenge problems will be on testing emerging SiSPI software technology research without the need for tailoring challenge problems to fit any particular acquisition program.

3.3.2.2. *Other SWTT Customers and Government Collaborators*

3.3.2.2.1. *ARL SiSPI BAA*

We have had a focus on monitoring the first software technology BAA from SiSPI, namely ARL BAA DAAD19-03-R-0017, Amendment 3, Research Area 1.15, Software Technologies Targeting Interoperability for Systems of Systems, April 2006. In this BAA, ARL has indicated that "The Software and Systems Test Track is one place prototype software could be delivered" from the ARL effort for experimentation with the BAA research products. We have viewed the ARL BAA as a sample of a SiSPI research thrust to be supported by SWTT, but not necessarily as defining the boundaries of SiSPI research. Accordingly, we have sought compatibility with the ARL BAA for our CONOPS and Architecture Framework, without narrowly focusing them on it.

Particular research focus areas for this ARL BAA, which would benefit from SWTT support, include the following (excerpted from the BAA Amendment 3):

a. Domain-specific modeling languages and semantics

b. Model-based design and development/engineering for system of systems architectures and ultra large scale software intensive architectures.

c. Models which support reflective (self-referential) capability

d. Principles and ontology development for organization of components, their design and construction

e. Verifiably correct generators and models

f. Re-engineering and Integration technologies (methods, tools, metrics, models, etc.) for legacy systems

TBD – As of the current writing of this Initial Submission of the Final Technical Report, the announcement of winners of this BAA, and their technologies which would be candidates for SWTT experimentation, have not been revealed by ARL pending required contractual processes. We are working closely with Mr. Glenn Racine of ARL to gather more detailed information on SWTT requirements from this BAA as they become available.

3.3.2.2.2. Other Government Collaborators

Herb Klumpe of the AFRL Information Directorate Information Grid Cyber Operations Branch (AFRL/IFGB) provided our team information on the AFRL Rome Research Site (RRS) networking infrastructure as it could relate to SWTT, including its current configuration and future plans. Mr. Klumpe briefed the processes and challenges associated with getting user accounts on the RRS computer systems and the software approval process that historically has taken weeks to months for approval. This interaction with Mr. Klumpe occurred at the SWTT Kickoff Meeting on 08 August 2006.

Our team interacted with Eddie Brooks of the DoD High Performance Computing Modernization Program (HPCMC), covering topics such as defense research engineering network (DREN) infrastructure and capabilities, access methods, and current user base. This interaction was part of a SWTT program telecon put together by our AFRL customer on 16 August 2006.

Other software technology programs, outside of the SiSPI umbrella, might benefit from the infrastructure and challenge problem set being developed for SWTT. Relationships such as these would provide momentum for SWTT and possibly additional funding sources for SWTT development and on-going operations and support. As a particular example, the Certification Techniques for Flight-Critical Systems (CerTA FCS) program from AFRL indicated a desire to consider use of SWTT as part of their program, including potential provision of resources for generating challenge problems particular to their research domain. These conversations occurred after the SWTT Mid-Term Review Meeting of 02 November 2006.

3.3.3. *Engaging Researchers*

3.3.3.1. *Researcher-Focused Survey*

During the first half of our Phase I activities, we developed and sent a survey to a distribution list of software technology researchers who might be likely to respond to emerging SiSPI software technology BAAs. The motivation of the survey was to collect information from the research community on potential utilization of SWTT to inform our CONOPS and Architecture work. The survey itself is shown in Section 5.1. Survey respondents filled in the survey fields in a Microsoft Word file and emailed the responses back to Dr. Doug Stuart at Boeing, who collected and summarized these responses.

The list of software technology researchers who were sent surveys was built from numerous sources, including:

- Members of the BSCW SWTT site
- Attendee list from the latest SiSPI Workshop at that time (17-19 May 2006, held in Arlington, VA)
- Attendee list from the SWTT Kickoff Meeting
- Distribution list used by OSD for SiSPI-related emails

The survey pulled information from the researchers in areas such as their likely technology contributions to solving software producibility problems (e.g., new software tools, new run-time technologies), their anticipated needs for challenge problems and other SWTT support, and their SWTT accessibility preferences (e.g., downloadable, hosting on a remote government site, executing on a remote publicly-available infrastructure like Emulab, etc.).

We also gained insight into anticipated researcher problem domains (Advanced electronics systems, Large scale embedded system of systems, etc.) and solution domains, such as Design and Analysis tools (architectural analysis, timing analysis, etc.) or Run-time infrastructure (middleware services, operating system schedulers, etc.).

A summary of researcher responses was put together and briefed at the Researcher-Focused Workshop and again at the SWTT Mid-Term Review. This summary is shown in Table II. In this table, affirmative responses to survey queries are shown in dark blue, lack of researcher interest is shown in light gray, and indications of possible researcher interest are shown in light blue.

Table II: Researcher-Focused Survey Response Summary

• **Technology Type to be Worked by Researcher:** • Development process tool/technology • Analysis tool/technology • Middleware technology • Networking technology/protocols • Code generator tool/technology • Meta code generation tool/technology • Other (Experimental techniques for tools, processes, technologies)
• **Technology Area to be Worked by Researcher** • Process improvement • Design time analysis • Run time analysis • Run time infrastructure • Other (Product Line Architecture)
• **Challenge Problem Domains of Interest to Researchers** • Advanced electronics systems (e.g., Advanced Mission Management Avionics Systems, Software-Defined Radio) • Large scale distributed embedded system-of-systems (e.g., FCS-like) • Large scale distributed real-time information exchange (e.g., GIG-like) • Next generation system of systems platform (advanced naval vessels, etc. e.g., DDG 1000-like) • Sensor networks

- **Challenge Problem Features of Interest to Researchers**
  - **System requirements**
  - **System architecture**
  - **System design**
  - **System model**
  - **System implementation**
  - **Component model**
  - **Component library**
  - **System/Component deployment information/files**
  - **Component implementations**
  - **Product line architecture**
  - **System timing requirements**
  - **Formal system specification**
  - **System development metrics**
  - **System deployment metrics**
  - **Other**

- **Targeted Development Context Envisioned by Researchers**
  - **Initial product development/creation only**
  - **Product upgrades/evolution only**
  - **Both**

- **SWTT Requirements Suggested by Researchers**
  - **Network *(wireless, Ethernet, Fibre Channel, CANbus, …)***
  - **Middleware *(CORBA, COM, TTA...)***
  - **Simulation environment *(network simulation,...)***
  - **Operating System *(Linux, Solaris, QNX, WinCE, ...)***
  - **Hardware *(embedded CPU, routers, sensors, ...)***
  - **Number/Type of elements *(100 CPUS, 10 sensors, 3 routers, ...)***
  - **Control flow paradigms *(Event driven, Periodic,...)***
  - **Data flow paradigms *(Pushing data, Pulling data,...)***
  - **Component structure/API *(Structural patterns, Configuration patterns, Distribution patterns. Does your work rely on a specific type(s) of application component?)***
  - **Thread location *(Internal to components? External to components? Both?)***
  - **Synchronization *(synchronous with inputs, with outputs, concurrency control...)***
  - **Scheduling protocols *(RMS, EDF, MUF...)***
  - **Special resource requirements, *(significantly large amounts of memory, throughput, communication bandwidth,...)***
  - **Other**

- **SWTT Access Options Preferred by Researchers**
    - **Do you have DREN access?**
  - **Would you:**
    - **Use SWTT at a remote site installed on SWTT-provided remote hardware, possibly available via DREN?**
    - **Use an SWTT downloaded and installed for use at your site?**
    - **Use an SWTT downloaded and installed for use at a third party site (e.g., Emulab)?**

- **Information Protection Preferences of Researchers**
  - **Have the capability to work with export controlled data?**
  - **Require the capability to work with export controlled data?**
  - **Have the capability to work with classified data?**
  - **Require the capability to work with classified data?**

- **SWTT Operator Training Requirements**
  - **What knowledge is needed by SWTT operators to enable them to understand/use your technology, or to support your use of the SWTT? Does your technology introduce new modeling notations, new architecture views, etc?**
    - Minimal training required

- **Integration Interfaces and Opportunities**
  - **How could your technology(s) be included in an integrated environment tested in the SWTT?**
    - Integrated into an OTIF-like entity such as what is currently implemented in ESCHER

After briefing this survey and results at the SWTT Mid-Term Review Meeting, OSD and AFRL requested that we make the information available to them. Dr. Doug Stuart of Boeing delivered the information to the government in a 07 November 2006 email.

3.3.3.2. Researcher-Focused Workshop

On Friday 20 October 2006 we held a researcher-focused workshop at Vanderbilt University in Nashville, Tennessee. The focus of the workshop was to interact with the software technology research community, to brief them on our approach to SWTT, and to get their feedback from a potential user's perspective. The Workshop was attended by OSD (Rob Gold), AFRL (Steve Drager, Bill McKeever), our Boeing-led SWTT team, and multiple members of the software research community. Participation by some was through a telecon and Webex set up by Boeing.

The agenda from the Workshop is shown in Figure 4. Our team started the day by briefing the assembled research community participants on SiSPI and the goals of SWTT, followed by a discussion on the classes of SWTT users that we envision. Then, an exposition of our CONOPS was presented, including our thoughts on challenge problems and use cases. This was followed by our SWTT architecture concepts. We then walked through in detail specific use cases involving SWTT utilization by a future SiSPI researcher, illustrated by UML diagrams and architecture diagrams (highlighting relevant architectural elements involved in each particular use case); this exposition of specific user interactions we termed "A Day in the Life of a SiSPI Researcher using the SWTT."

- **SWTT Goals**
- **Classes of Users**
  - **SiSPI Research Community Members**
    - **Focus of Today**
  - **SWTT Staff**
  - **Government Program Offices**
  - **Application Developers**
- **SWTT CONOPS**
  - **Concepts for research community use of SWTT**
- **SWTT Architecture**
  - **Architectural concepts supporting researcher utilization and support for R&D collaboration**
- **Day in the life …**
- **Gather Input on How the SWTT Can Support Researchers**
  - **Research Technology Domains**
  - **Research Technology Infrastructure Support Needs**

Figure 4: Researcher-Focused Workshop Agenda

Following "A Day in the life ...", we reviewed the content of our Researcher-Focused survey that was sent out prior to the Workshop and looked at a summary of responses that had been received thus far. We then opened the session up to feedback from the research community members.

The most beneficial element of the workshop for our SWTT team was the feedback we gathered from the researcher community. In general, researcher comments focused on SWTT usability. For example, we need to articulate clearly how their research-under-test is integrated into SWTT for experiments. Also, it became clear that in our design discussion, we need to clearly distinguish research-under-test from challenge problem components and Test Track infrastructure (e.g., metrics collection functions).

There was also significant discussion on the concept for enhancing tool transitionabilitiy by supporting integration into feature-rich tool chains. To support this integration, we recommend the concept first developed on other DoD research programs, wherein developers of tools formally specify their tool interfaces and semantics. It became evident in the Workshop, though, that some researchers may prefer simple testing of tools without the need to formally specify these interfaces and semantics. Our SWTT will accommodate this non-integrating approach if desired by researcher and their research customer.

3.3.3.3. Wiki Site

Early in the Phase I effort a Wiki site for future open R&D collaboration, accessible on the public Internet, was stood up on the Vanderbilt ISIS web servers. At this early stage in SWTT evolution, the Wiki contains general information about the current study effort. In Phase II of SWTT development, the Wiki would evolve into a full collaborative environment allowing for researcher, SWTT staff, and government stakeholder interaction. The Wiki (https://wiki.isis.vanderbilt.edu/swtt) leverages HTTP with Socket Secure Layer (SSL), for encrypting content before transmitting over the Internet.

3.3.4. Engaging Industry

The industry members of our team, Boeing and Raytheon, have socialized the SWTT effort within our companies. The purpose of these interactions is multifold:

- to foster future transition of SiSPI technologies that will have their worth illustrated in SWTT; contractor use of SiSPI tools and run-time technologies on DoD programs can be encouraged with successful SWTT demonstration coupled with adequate visibility of these successes within the contractor community

- to encourage future participation by contractor program teams to contribute to challenge problem definition and to infrastructure development, resulting in challenge problem realism and availability of off-the-shelf infrastructure elements (e.g., existing application middleware, other Open Experimental Platform artifacts, etc.) that can be integrated into SWTT

- to make program teams aware of the benefits of the future SWTT, potentially resulting in expanded interest, use, and investment in the Test Track spurred by interest shown by contractors and their DoD customers

Particular activities by Boeing and Raytheon have included briefings on the SWTT concept by our SWTT personnel at company-wide meetings of software technologists and company tools groups. Boeing and Raytheon SWTT personnel have also had focused sessions with particular programs, such as (1) manned and unmanned aircraft, Joint Tactical Radio System, Future Combat Systems, etc., at Boeing; and (2) DDG 1000 and Precision Weapons programs (Non-Line-of-Sight Launch System, Small Diameter Bomb II, Mid Range Munition, and Exoatmospheric Kill Vehicle) at Raytheon

3.3.5. Trades

A focus of our Phase I effort is to leverage off-the-shelf infrastructure and assets as much as possible to enable affordable development of the SWTT. This resulted in a number of trades looking at what new capabilities need to be developed versus what leveragable capabilities already exist. We also considered what tailoring of existing infrastructure and assets may be needed to craft an integrated SWTT. Existing infrastructure and other assets targeted included the following:

- Open Experimental Platforms (OEPs) and Challenge problems – a rich history of OEP work exists to pull from among our extended team. From Boeing, the DARPA MoBIES, PCES, and SEC programs all involved generation of open challenge problems and OEPs for software research experimentation and evaluation. From Raytheon, the DARPA ARMS program brings similar artifacts, in a program that also included Boeing involvement. An emerging program from AFOSR, "Human Centric Design Environments for Command and Control Systems: The C2 Wind Tunnel.", being worked by Vanderbilt University, is generating system of systems challenge problems involving multiple moving and fixed entities including human-operated command and control (C2) nodes.

- System of Systems Simulation Infrastructure – A number of existing simulation infrastructures have been considered from work done by our extended SWTT team and other organizations that we have collaborated with. This includes multi-entity simulation work done on the DARPA SEC and MICA programs by Boeing, as well as infrastructures used by the Boeing simulation technology organization. Work being done by Vanderbilt on the AFOSR C2 Wind Tunnel project has also been considered, including both an entity simulation infrastructure and a network simulation. In addition, we have interacted with another Boeing AFRL customer on consideration of the FLexible Analysis Modeling and Exercise System (FLAMES), a framework for developing constructive simulations and interfaces between constructive, virtual, and live simulations. Other simulation infrastructures considered include High Level Architecture (HLA), Distributed Interactive Simulation (DIS), Integrated Collaborative Environment (ICE), and OMNeT.

- Application Middleware Infrastructure – Application middleware that we can pull from in building experimental platforms and challenge problems for SiSPI research includes the following library of work that has been done by Boeing, Raytheon, and others – ACE/TAO, RT-CORBA, DDS, SOAP, Product line Real-time Software component model (PRiSm) from the DARPA MoBIES and PCES program, PRiSm-Java from the DARPA PCES program, Open Control Platform from the DARPA SEC program, etc.

- Physical Hosting and Experimentation Infrastructure – Here our focus is again on leveraging existing infrastructure, at least in initial phases of SWTT development, for cost effectiveness. We have considered use of ESCHER servers; servers within Boeing and Raytheon with secure access to the public Internet; DREN access to the Rome Research Site and other DREN-accessible options; the Vanderbilt Emulab for executing experiments on representative hardware, operating systems, and networks; and the Utah Emulab for executing experiments on representative hardware, operating systems, and networks. Because of anticipated challenges in hosting researcher software and experiments on RRS and other DREN assets, an RRS instantiation for SWTT might consist of data and results repositories, with more open systems such as ESCHER and Emulab being used for on-going researcher experimentation. Our architectural concept of a downloadable Test Track also allows for experimentation without the need to load research software on an Air Force computer. As part of a study of DREN applicability to SWTT, we undertook an effort to identify a list of government resources, both hardware and software, available via the High Performance Computing initiative. The results of this effort are shown in Section 5.2. It is anticipated that there may not be a need to procure new hardware for ESCHER or the Emulabs to support SWTT, although a usage fee may be appropriate to support expanded ESCHER and Emulab use due to SWTT activities. Furthermore, ESCHER infrastructure has an established tool chain integration framework that can be leveraged in our architecture.

3.3.6. CONOPS and Architecture Development and Documentation

All other program activities discussed so far, including collaborations, researcher engagement, and trades, were all aimed at the major focus of the program – development and documentation of SWTT CONOPS and Architecture. This section of the Final Technical Report provides background information on our approach to CONOPS and Architecture development, while the results of our Phase I CONOPS and Architecture definition work are documented in our Final Review briefing charts and two companion CDRL documents:

- Concept of Operations for the Software and Systems Test Track, AFRL Contract Number FA8750-06-C-0213, CLIN 0002, CDRL Data Item No. A003
- Architecture Framework for the Software and Systems Test Track, AFRL Contract Number FA8750-06-C-0213, CLIN 0002, CDRL Data Item No. A004

3.3.6.1. CONOPS

Development of the CONOPS, which describes SWTT characteristics from a user's point of view, started with identification of the classes of SWTT users that we envision. We then elaborated use case families for these classes of users. Finally, we developed initial definitions of Challenge Problems and Application Domains.

The SWTT user classes defined include:

- Software Technology Researchers, who will be testing their research results and products on SWTT
- Government Program Offices who (1) are funding software technology work and want to use SWTT to explore performance of those technologies, or who (2) have acquisition programs and want to use SWTT to test the utility of key tools, potentially to help them with unsolved problems
- Industrial Users who represent some of the ultimate consumers of SiSPI produced technology, and who may use the SWTT to test the utility of SiSPI research products, or supply or validate the challenge problems that guide and validate that research
- SWTT Staff Members, who will manage and maintain the Test Track

Various use case families were identified, including Configuring the SWTT, Modifying the SWTT, Experimenting using the SWTT, Coordinating with the SWTT, Collaborating via the SWTT, Training on the SWTT, and Mining the SWTT. More details on these use case families can be found in our CONOPS document. We have made extensive use of UML diagrams as part of our use case documentation, in both our CONOPS document and our program review briefing charts. An example diagram is shown in Figure 5.

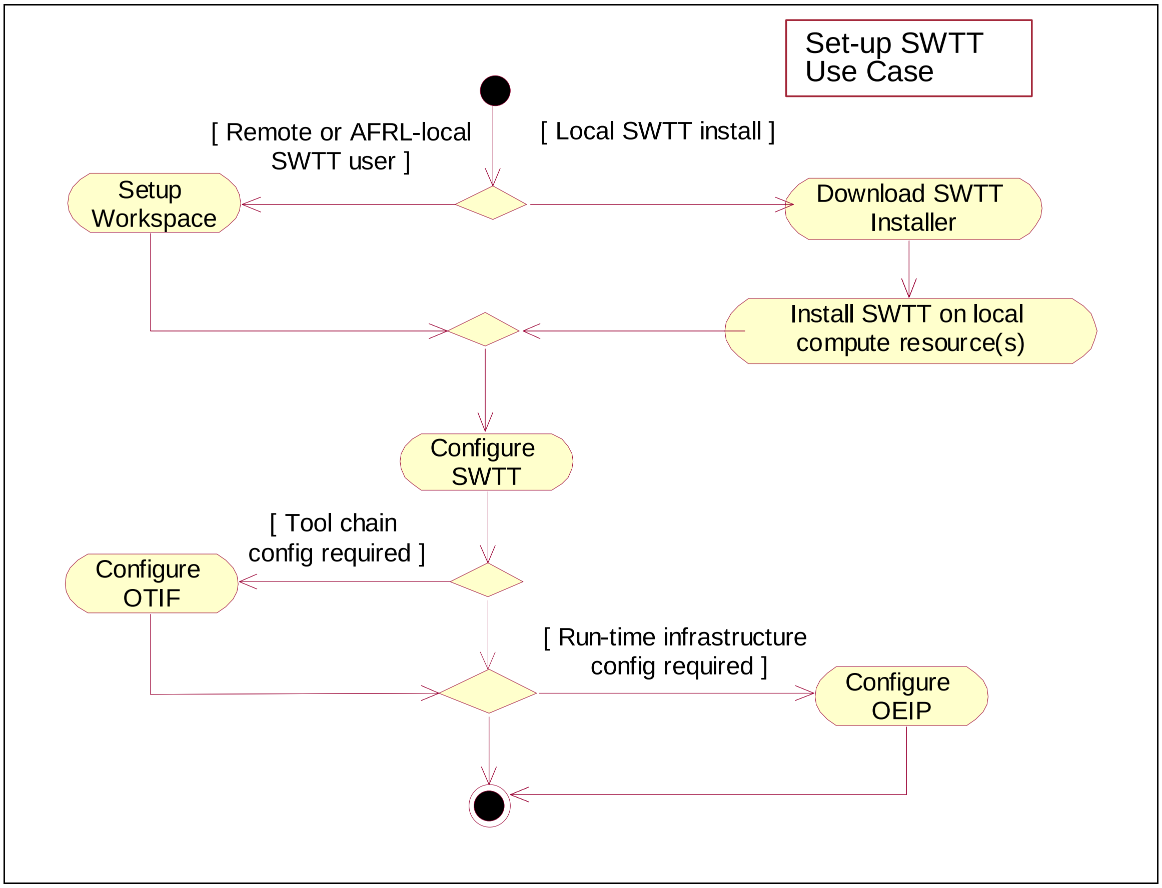

Figure 5: UML Diagram for the "Configuring SWTT" Use Case

Challenge problems and application domain ideas were pulled from the rich set of applications and past work in OEPs that our extended team members have been involved in.

From a process standpoint, the format and content of the CONOPS document is based on IEEE Std 1362-1998, *IEEE Guide for Information Technology – System Definition – Concept of Operations (ConOps) Document*. The SWTT BAA directive to have the CONOPS be a "user-oriented document that describes system characteristics for a proposed system from the users' viewpoint" was very well aligned with the intent of the IEEE standard, which used identical language to describe the goals of the documetation.

3.3.6.2. Architecture

For Architecture definition, we identified the software and hardware elements of SWTT that would enable analyzing the effectiveness of software technologies on representative software intensive systems, in addition to identifying relevant organization aspects. This led to the identification of three major aspects to the SWTT architecture, and we worked to mature those aspects:

- **Technical Architecture –** The Technical Architecture defines the infrastructure and technical components that enable the Test Track capabilities. It provides the foundation on which the SWTT is developed to address DoD-strength challenge problems and support various challenge problem environments. In particular, the Technical Architecture will be built on top of an Open Tool Integration Framework (OTIF) and an Open Experiment Integration Platform (OEIP). The OTIF provides an open software development infrastructure and hooks for various tools for linking them into toolchains to form specific integrated solutions. The OEIP provides the overall framework for integrating, testing and evaluating various types (and complexities) of tools and run-time technologies.
- **Organizational Architecture –** The Organizational Architecture presents a model for how the SWTT will operate. Specifically, this defines an operational structure and a set of process and procedures that will ensure the SWTT meets the CONOPS and supports the various users, use cases, challenge problems, and challenge problem environments.
- **Deployment Architecture –** The Deployment Architecture addresses the way in which the SWTT will be implemented. This will be done in conjunction with the CONOPS development for consistency and to ensure that the deployment methodologies support the SWTT operational needs. The Deployment Architecture also addresses deployment considerations and issues associated with the implementation approaches.

The OTIF element of the Technical Architecture has been patterned after a similar construct within ESCHER, with its notion of tool chains enabled by innovative tool integration concepts. OEIP borrows from multiple infrastructure elements that were explored during our look at existing OEPs, Challenge Problems, System of Systems Simulation Infrastructures, Application Middleware Infrastructures, and Physical Hosting and Experimentation Infrastructures.

4.0 Summary

4.1. Conclusions

The SWTT Phase I effort has resulted in a CONOPS and Architecture definition that will result in a valuable asset for testing emerging software technology. The SiSPI, and its funded technology programs, will be the main beneficiary of this work. Also, interest expressed by other programs will help maintain momentum for SWTT and may also result in additional funding for SWTT development and support.

Our team has the right balance of industry, academic, and consortium involvement. Our industry component, Boeing and Raytheon, has a rich history in developing OEPs and challenge problems, and executing and evaluating software technology research. Artifacts from these programs and our past experience are well aligned with SWTT. Our academic partners, Vanderbilt and UC-Berkeley, are involved in many pursuits that are on the leading edge of software technology, including run-time technologies and rich tool environments. The ESCHER consortium exists today as an honest broker for research contributions and government / industry exploitation of that research, including existing infrastructure for data repositories, tool chains, and researcher collaboration mechanisms.

During execution of the Phase I effort, we made extensive use of multiple methods (Webex, SharePoint, etc.) to coordinate activities from our talented, but dispersed team. Frequent communication with our customers was also very valuable, especially the continued guidance from William McKeever and Steven Drager of AFRL. Our customers were proactive in arranging meetings and conversations with other government entities that could affect eventual SWTT implementation.

Interaction with and feedback from the research community was invaluable. The use of the Researcher-Focused Surveys and Workshop was extremely effective in soliciting ideas that fed our CONOPS and Architecture. Our aim is to ensure that the CONOPS and Architecture appropriately reflect the needs of this research community, as well as the customers funding their research.

We have identified numerous sources of leveragable and off-the-shelf components that will lead to an affordable and scalable SWTT. Our team has in most cases developed or has extensive experience in use of these components.

We look forward to the opportunity to build the SWTT, an entity that has the potential to have a wide scope of beneficiaries, potentially reaching across the US Air Force, Army, and Navy.

5.0 Attachments

5.1. Researcher-Focused Survey

SWTT SISPI Researcher Technology Survey

This survey is intended to elicit information, from researchers developing software tools and technology who might pursue funding through the emerging OSD Software-Intensive Systems Producibility Initiative (SISPI), for the purposes of characterizing candidate requirements for the Software Test Track (SWTT).

Many of the questions present a number of options. Select any and all that apply to any or all of the technologies that you may be bringing to the SWTT. Also, feel free to elaborate on any of your responses, and to include additional comments.

Please email survey responses to Dr. Doug Stuart (douglas.a.stuart@boeing.com) by Friday 13 October 2006. Survey results will be discussed in a virtual and face-to-face Workshop to be held at Vanderbilt University on Friday 20 October 2006 with participation enabled for remote participants via telecon and Webex.

SISPI Researcher Top-Level Information	
Author (Last, First MI):	Organization:
E-Mail Address:	Phone Number:

1. Description of your potential SISPI Research: *(Provide a brief description of your research.)*

2. Technology Type: *(Indicate the type(s) of technology(s) you are developing. Mark with an X all that apply.)*

_____Development process tool/technology
_____Analysis tool/technology
_____Middleware technology

_____Networking technology/protocols
_____Code generator tool/technology
_____Meta code generation tool/technology
_____Other (please explain)____

Comments:

3. Technology Area: *(List the Technology Area(s) your research addresses for each technology, if necessary. Mark with an X all that apply.)*

_____Process improvement
_____Design time analysis
_____Run time analysis
_____Run time infrastructure
_____Other (please explain)____

Comments:

4. Challenge Problems: *The SWTT is intended to support testing of SISPI research products that will enable affordable development of software for large-scale, complex, embedded and net-centric systems. The test track will include industrial-strength challenge problems that are representative of real software-intensive systems. These challenge problems will be used to stimulate technology development, serve as test cases for emerging technologies, and assist in maturing developing technologies.*

4a. Challenge Problem Domains: *Identify those challenge problems (or challenge problem domains) to which your technology(s) is (are) applicable. Place an X in the left column for all that apply.*

Applies?	**Challenge Problem / Challenge Problem Domain**
	Advanced electronics systems (e.g., Advanced Mission Management Avionics Systems, Software-Defined Radio)

	Large scale distributed embedded system of systems (e.g., FCS-like)
	Large scale distributed real-time information exchange (e.g., GIG-like)
	Next generation system of systems platform (advanced naval vessels, etc. e.g., DDG 1000-like)
	Sensor networks

4b. Challenge Problem Features: *Identify those features of challenge problems (or challenge problem domains) that are required for your technology to be applicable. For example, doing model level consistency checking would require a system model. Place an X in the left column for all that apply.*

Applies?	**Feature of Challenge Problem / Challenge Problem Domain**
	System requirements
	System architecture
	System design
	System model
	System implementation
	Component model
	Component library
	System/Component deployment information/files
	Component implementations
	Product line architecture
	System timing requirements
	Formal system specification
	System development metrics

	System deployment metrics
	Other challenge problem elements (please explain):

5. Targeted Development Context: *(Indicate with an X in what program context each research product is expected to be used.)*

_____Initial product development/creation
_____Product upgrades/evolution
_____Both

Comments (Indicate the ways in which your technologies are particularly relevant to particular parts of the product life cycle):

6. SWTT Requirements: *(Provide a description of the assumptions your product is dependent upon, if any. Are there specific SWTT capabilities that will be required to support your work? Place an X in the left column for all that apply, and provide any additional detail.)*

____Network *(wireless, Ethernet, Fibre Channel, CANbus, ...)***, specifically:**

____Middleware *(CORBA, COM, TTA...)***, specifically:**

____Simulation environment*(network simulation, application level environment simulation, ...)***, specifically:**

____Operating System *(Linux, Solaris, QNX, WinCE, ...)***, specifically:**

____Hardware *(embedded CPU, routers, sensors, ...)***, specifically:**

____Number/Type of elements *(100 CPUS, 10 sensors, 3 routers, ...)***, specifically:**

____Control flow paradigms *(Event driven, Periodic,...)***, specifically:**

____**Data flow paradigms** *(Pushing data from suppliers to consumers, Pulling data from suppliers by consumers,...)***, specifically:**

_____**Component structure/API** *(Structural patterns, Configuration patterns, Distribution patterns. Does your work rely on a specific type(s) of application component?)***, specifically:**

____**Thread location** *(Internal to application components only? External to application components only? Both?)***, specifically:**

____**Synchronization,** *(synchronous with inputs, with outputs, concurrency control...)* **specifically:**

____**Scheduling protocols** *(RMS, EDF, MUF...)***, specifically:**

____**Special resource requirements,** *(significantly large amounts of memory, throughput, communication bandwidth,...)* **specifically:**

____**Other (please explain)____**

7. SWTT Access options: *How do you anticipate accessing/using the test track (Indicate with Y / N in left column)?*

_____**Do you have DREN (Defense Research and Engineering Network) access? (see http://www.hpcmo.hpc.mil/Htdocs/DREN/index.html)**

_____**Use SWTT at a remote site installed on SWTT-provided remote hardware, possibly available via DREN?**

_____**Use an SWTT downloaded and installed for use at your site?**

_____ **Use an SWTT downloaded and installed for use at a third party site (e.g.,**

Emulab)?

8. Information protection. *What are your anticipated information protection needs/capabilities when using the SWTT? (Indicate with Y / N in left column)*

_____Do you have the capability to work with export controlled data?

_____Do you require the capability to work with export controlled data?

_____Do you have the capability to work with classified data?

_____Do you require the capability to work with classified data?

9. SWTT Operator Training Requirements: *(What knowledge is needed by SWTT operators to enable them to understand/use your technology, or to support your use of the SWTT? Does your technology introduce new modeling notations, new architecture views, etc?)*

10. Integration Interfaces and Opportunities: *(How could your technology(s) be included in an integrated environment tested in the SWTT? For example, a code instrumentation tool could be integrated as an Eclipse plug-in, or a tool for performing consistency checks on a system architectural model could be integrated into a development tool chain via well defined model interchange formats (e.g. MOF, XMI) exchanged using a backplane such as the ESCHER (www.escherinstitute.org) Open Tool Integration Framework.)*

11. Comments: *(Use this space for any other comments you may have on your potential use of the SWTT, or other ways in which the SWTT could be made a valuable resource for SISPI and for the development of software intensive DOD systems.)*

5.2. DREN Study

DREN Overview

The Defense Research and Engineering Network (DREN) is DoD's recognized research and engineering network. The DREN is a robust, high-capacity, low-latency nation-wide network that provides connectivity between and among the HPCMP's geographically dispersed High Performance Computing (HPC) user sites, HPC Centers, and other networks. The DREN Wide Area Networking (WAN) capability is provided under a commercial contract. The DREN WAN service provider has built DREN as a virtual private network based on its commercial infrastructure.

The DREN provides digital, imaging, video, and audio data transfer services between defined service delivery points (SDPs). SDPs are specified in terms of WAN bandwidth access, supported network protocols [Multi Protocol Label Switching, Internet Protocol (IP), Asynchronous Transfer Mode (ATM)], and local connection interfaces. DREN currently supports both IP version 4 (IPv4) and IP version 6 (IPv6) at bandwidths from DS-3 (45 Mbps) at user sites up to OC-48c (2.488Gbps) at selected HPC Centers. Future bandwidths will scale even higher. Expansions or enhancements to the DREN as a whole are accomplished through the addition of defined SDPs or modifications to the operating specifications of existing SDPs. The sites connected by DREN services may be at virtually any location in the continental United States, including Alaska and Hawaii, and at OCONUS sites.

Incorporating the best operational capabilities of both the DoD and the commercial telecommunications infrastructure, DREN is the official DoD long-haul network for computational scientific research, engineering, and testing in support of DoD's S&T and T&E communities. It has also been designated as a DoD IPv6 pilot network by the Assistant Secretary of Defense (Networks & Information Integration)/DoD Chief Information Officer [ASD (NII)/DoD CIO]. DREN enables over 4,300 scientists and engineers at DoD and other government laboratories, test centers, universities, and industrial locations to use HPCMP computing resources. Since its inception, DREN has been very active in transferring leading edge network and security technologies across DoD and other federal agencies. Since users and resources are scattered throughout the United States, strong interconnectivity with other major networks and high performance test beds at key interconnect points are critical for optimal use of DoD HPC resources.

HPCMP Baseline Configuration Overview

Baseline Configuration is a DoD High Performance Computing Modernization Program (HPCMP) Project tasked to define a common set of capabilities and functions so that users can work more productively and collaboratively when using the HPC resources at multiple computing centers.

HPCMP Participating Sites

Army Research Laboratory (ARL), Aberdeen Proving Ground, MD
Arctic Region Supercomputing Center (ARSC), Fairbanks, AK
Aeronautical Systems Center (ASC), Wright Patterson AFB, OH
Army Engineer Research and Development Center (ERDC), Vicksburg, MS
Maui High Performance Computing Center (MHPCC), Kihei, HI
Naval Oceanographic Office (NAVO), John C. Stennis Space Center, MS

Table III: DREN Hardware Summary – Multiple Sites

Site*	Mfg	Model	Decommission	OS	CPU Type	CPU Speed	Avg Capability*	# of CPUs
ARL	IBM	P-Series 690 SP	9/30/2006	AIX 5.2	Power4+	6.8 GFLOPS (1.7 GHz)	6.55	128
ARL	Linux Networx	Evolocity II	9/30/2007	Red Hat Enterprise AS 4.0	IA-32	6.12 GFLOPS (3.06) GHz	5.21	256
ARL	Linux Networx	Evolocity II	9/30/2008	SLES 9	Intel Xeon EM64T	7.2 GFLOPS (3.6 GHz)	5.80	2048
ARL	SGI	Onyx2 IR2		IRIX 6.5	R14000	1 GFLOP (500 MHz)		16
ARL	SGI	Onyx2 IR3		IRIX 6.5	R14000	1 GFLOP (500 MHz)		16
ARL	SGI	Onyx 3000		IRIX 6.5	R14000	1 GFLOP (500 MHz)		8
ARL	SGI	Onyx 3000 IR3		IRIX 6.5	R14000	1 GFLOP (500 MHz)		24
ARL	IBM	Cluster 1350	9/30/2008	SLES 9	AMD Opteron	4.4 GFLOPS (2.2 GHz)		32
ARL	Linux Networx	Evolocity II	9/30/2008	SLES 9	Intel Xeon EM64T	7.2 GFLOPS (3.6 GHz)		32
ARL	SGI	Altix 3000	9/30/2008	SGI Propack 3	Intel Itanium2	3 GFLOPS (1.5 GHz)		16
ARL	IBM	Cluster 1350	9/30/2008	SLES 9	AMD Opteron	4.4 GLOPS (2.2 GHz)	4.73	2304
ARL	SGI	Onyx2 IR3		IRIX 6.5	R10000	500 MFLOPS (250 MHz)		16
ARSC	Cray	X1		UNICOS/mp 2.5.23	Cray X1	12.8 GFLOPS per MSP (800 MHz) / 3.2 GFLOPs per SSP (800 MHz)	HABU Pending	128 MSP (512 SSP)
ARSC	IBM	P4		AIX 5.2	IBM Power4	5.2 - 6.8 GFLOPS per CPU (1.3 - 1.7 GHz)	HABU Pending	800
ASC	HP	XC Cluster		Linux RedHat 4.0 EL	AMD Opteron	5.6 GFLOPS (2.8 GHz)		2048
ASC	SGI	Altix 3700 BX2		Linux RedHat 2.4	Intel Itanium 2	6.4 GFLOPS (1.6 GHz)		2048
ASC	SGI	Origin 3900		IRIX UNIX	MIPS R16000	1.4 GFLOPS (700MHz)	3.08	2048
ASC	Compaq	SC-45	9/30/2006	True64 UNIX w/ SC (Sierra Cluster) 2.6	AlphaEV6.8	2 GFLOPS (1 GHz)	2.74	836
ERDC	Cray	X1	09/30/2006	UNICOS/mp 3.0.23	Cray X1	12.8 GFLOPS per MSP (800 MHz) / 3.2 GFLOPS per SSP (800 MHz)	5.65	64 MSP (256 SSP)
ERDC	Cray	XT3	01/31/2010	UNICOS/ls 1.2.x	Opteron	5.2 GFLOPS (2.6 GHz)	?	4176
ERDC	Compaq	SC45	09/30/2006	TRU64 V5.1	Alpha 21264	2 GFLOPS (1000 MHz)	2.74	512
ERDC	SGI	Origin 3900	06/30/2007	IRIX64 6.5	MIPS R16000	1.4 GFLOPS (700 MHz)	3.08	1024
ERDC	SGI	Onyx 340	09/30/2006	IRIX64 6.5	MIPS R14000	1.2 GFLOPS (600 MHz)	NA	32
ERDC	SGI	Origin 2000	09/30/2007	IRIX64 6.5	MIPS R10000	390 MFLOPS (195 MHz)	NA	32
MHPCC	IBM	NH-2 SMP P3		AIX 5.1	Power3	1.5 GFLOPS (375 MHz)	1.45	736
MHPCC	IBM	SMP-P4		AIX 5.1	Power4	5.2 GFLOPS (1.3 GHz)	2.58	320
NAVO	IBM	P4	12/31/2006	AIX 5.2L	IBM Power4	5.2 Gflops (1.3 GHz)	2.10	1408
NAVO	IBM	P4+	09/30/2008	AIX 5.2L	IBM Power4+	6.8 Gflops (1.7 GHz)	6.55	512
NAVO	IBM	P4+	09/30/2008	AIX 5.2L	IBM Power4+	6.8 Gflops (1.7 GHz)	6.55	2944
NAVO	Graphstream	Gen 3		RedHat Enterprise WS 3.0	AMD Opteron	4.4 Gflops (2.0 GHz)	N/A	20
NAVO	Graphstream	Gen 3		RedHat Enterprise WS 3.0	AMD Opteron	4.4 Gflops (2.0 GHz)	N/A	10

*** Definitions**

ARL = Army Research Laboratory Major Shared Resource Center
ARSC = Arctic Region Supercomputing Center
ASC = Aeronautical Systems Center Major Shared Resource Center
ERDC = Engineer Research and Development Center Major Shared Resource Center
MHPCC = Maui High Performance Computing Center
NAVO = Naval Oceanographic Office Major Shared Resource Center

HPC = High Performance Computer
TDS = Test Development System
SciVis = Scientific Visualization
Average Capability = Size dependent, average benchmark capability: The number of Habu-equivalents represented by the entire system

Table IV: DREN Software Summary – ARL

Site	Name	Vendor	IBM Cluster 1350	IBM p690 SP	LNXI Evolocity II	SGI Altix
ARL	Abaqus	HKS	6.5-2	6.5-4	6.6-1	6.5-1
ARL	Accelrys	Accelrys			1.4	
ARL	ADF	SCM		2005.01b	2006.01	
ARL	ALE3D	LLNL			4.2.1	
ARL	ANSYS	ANSYS, Inc.		10.0	10.0	
ARL	AVS Suite	AVS				
ARL	BRL-CAD	Surviac ASO	7.8.0			
ARL	CASTEP	Accelrys			1.4	
ARL	CAVELib	VRCO				
ARL	Cerius	Accelrys			4.10L	
ARL	CFD++	Metacomp Technologies	5.1.2	5.1.2	5.1.2	5.1.1
ARL	CFX	ANSYS, Inc.		5.7.1	10	
ARL	Chemkin	Reaction Design		4.0.2	4.0.2	
ARL	Chimera Grid Tools	NASA AMES			1.9	
ARL	Cobalt	Air Force Research Lab			60	
ARL	Crystal	University of Torino		98		
ARL	CTH	Sandia National Labs	7.1	7.1	7.1	interim2003
ARL	Cubit	Sandia National Labs	8.0.1		10.0	9.1
ARL	Dmol3	Accelrys			1.4	
ARL	Ensight	CEI	8.0.5	8.0.5	8.0.5	8.0.5
ARL	EPIC	AHPCRC			2003	
ARL	eXceed					
ARL	FAST	NASA AMES				
ARL	FieldView	Intelligent-Light				
ARL	Fluent	Fluent	6.2.21	6.3.13	6.3.13	6.2.16
ARL	Gambit	Fluent	2.2.30	2.2.30	2.2.30	
ARL	Gamess	Ames Laboratory		jul05	jul05	
ARL	GASP	Aerosoft	4.2.1		4.2.2	
ARL	Gaussian Suite	Gaussian	2003d01	2003d01	2003d01	2003c02
ARL	Gaussview	Gaussian	3.09	3.07	3.09	
ARL	Gridgen	Pointwise		15	15	
ARL	Hypermesh	Altair Engineering		7.0	7.0	
ARL	ICEM CFD	ANSYS, Inc.		10.0	10.0	
ARL	Imagine	Leica Geosystems				
ARL	ISIGHT	Engineous Software		9.0	9.0	
ARL	Lightwave	NewTek				
ARL	LS-Dyna	LSTC	970	971	970	970
ARL	Mathematica	Wolfram Research		5.0	5.2	
ARL	MATLAB	Mathworks	7.2.0	6.0	7.2.0	
ARL	Maya	Autodesk			7.0	
ARL	Mesodyn	Accelrys			1.4	
ARL	MuSES	Thermoanalytics, Inc.			7.1.3	
ARL	NAG Libraries	NAG				
ARL	Ncargraphics	NCAR Graphics Group		4.2.3	4.4.1	
ARL	netCDF	Unidata	3.6.1	3.5.1	3.6.0	3.5.1
ARL	Opari	Forschungszentrum Julich		1.1		
ARL	Overflow	NASA AMES	2.0y	2.0y	2.0y	
ARL	Pandemonium	XAOS Tools				
ARL	PAPI	Innovative Computing Lab		3.2.1	3.2.1	
ARL	Parallel Virtual Machine	Oak Ridge National Lab				
ARL	Patran	MSC		2003R2	2005r2	
ARL	Pegasus	NASA AMES			5.1	
ARL	PETSc	Argonne National Labs	2.1.5	2.3.0	2.3.0	
ARL	ProEngineer	PTC			2.0	
ARL	PST	Colorado Research Assn.				
ARL	RenderMan	Pixar				
ARL	SAF	SAIC			3.1	
ARL	SAMUEL	NRL-NCARAI			1-JUL-97	
ARL	SPRNG	Florida State University		1.0	2.0	
ARL	TAU	University of Oregon		2.12.7	2.14.2	
ARL	Tecplot	Amtec	10	360	360	10
ARL	Tempus	HyPerComp				
ARL	TotalView	Etnus		7.0	7.2	6.4
ARL	TrueGrid	XYZ Scientific App Inc	3.6.8	3.6.8	3.6.8	
ARL	Vis5d	Vis5d+ Project				
ARL	Wind	NPARC			5	
ARL	Xpatch	SAIC	4.7.22		4.7.22	

Table V: DREN Software Summary – ARSC

Site	Name	Vendor	IBM P4/1.7	Cray X1
ARSC	AIX	IBM	5.2	
ARSC	bison	GNU	1.35	1.875
ARSC	C	vendor	6.0.0.6	5.2.0.0
ARSC	C	GNU	3.3.1	
ARSC	C++	vendor	6.0.0.0	5.2.0.0
ARSC	C++	GNU	3.3.1	
ARSC	CRAYLibsci	Cray, Inc.		5.0.0.3
ARSC	CVS	Concurrent Versions System	1.11.1p1	1.11.5
ARSC	EMACS	GNU/Free Software Foundtion	21.2.1	
ARSC	ESSL	IBM	4.1.0.0	
ARSC	Fortran	GNU	3.3.1	
ARSC	Fortran 90	vendor	8.1.1.4	5.2.0.0
ARSC	GNU make	GNU	3.80	3.79.1
ARSC	GNU tar	GNU	1.13	1.13
ARSC	gzip	GNU	1.2.4	1.2.4
ARSC	HDF	NCSA	4.1r5	
ARSC	HPM Toolkit	IBM	252	
ARSC	less	GNU		381
ARSC	Matlab	Mathworks	6.5	
ARSC	MPT	Cray, Inc.		2.3.0.4
ARSC	NCAR Graphics	NCAR	4.3.0	
ARSC	NCL	NCAR	4.2.0.a030	
ARSC	nedit	www.nedit.org	5.3	
ARSC	NETCDF	Unidata	3.5.0	3.5.0
ARSC	Netscape	Netscape Comm. Corp	4.79	
ARSC	PAPI	ICL @U.TN.	2.3.4.3	
ARSC	PBS Pro	Altair Engineering		5.3.4c
ARSC	perl	www.perl.com	5.8.0	5.6.1
ARSC	PESSL	vendor	3.1.0.1	
ARSC	POE	IBM	4.1.0.3	
ARSC	TCL/TK	Scriptics	8.3.3	8.3.4
ARSC	tcsh	www.tcsh.org	6.11.00	6.12.00
ARSC	Totalview	Cray/Etnus	6.4.0.0	6.3.0.1
ARSC	TurboMP	IBM	3.0.1	
ARSC	UNICOS/mp	Cray, Inc.		2.5.19
ARSC	VAMPIRtrace	Pallas	4.0	

Table VI: DREN Software Summary – ASC

Site	Name	Vendor	SGI Origin 3900	COMPAQ SC-45	SGI Altix	HP XC
ASC	ABAQUS	HKS	6.5-6	6.5-3	6.6	6.6
ASC	ACAD	Lockheed Martin				
ASC	ACES II	Univ of Florida	2.4			
ASC	AMSOL	University of Minnesota		6.6		
ASC	ANSYS	ANSYS	10.0	10.0	10.0	
ASC	archive	PSTOOLKIT ORG	2003 Sept 17	2003 Sept 17	2003 Sept 17	
ASC	aspell	Free Software Foundation	0.60.4	0.60.4	0.60.4	0.60.4
ASC	ATK	Free Software Foundation	1.9.0			
ASC	autoconf	Free Software Foundation	2.59	2.59	2.59	2.59
ASC	automake	Free Software Foundation	1.9.6	1.9.6	1.9.6	1.9.6
ASC	AVS	Advanced Visual Systems				
ASC	AVS/EXPRESS	Advanced Visual Systems				
ASC	AVUS	AFRL/VAAC	01Jan2004	01Jan2004	01Jan2004	01Jan2004
ASC	Bash	Free Software Foundation	3.0	3.0		3.0
ASC	Berkeley Unified Parallel C	UC Berkley			2.2.1	
ASC	Bison	Free Software Foundation	2.1	2.1	2.1	2.1
ASC	bzip2	Julian Seward	1.0.3	1.0.3	1.0.3	
ASC	C	SGI	7.4.3			
ASC	C	Compaq		6.4-014		
ASC	C	Intel			8.1	
ASC	C++	SGI	7.4.3			
ASC	C++	Compaq		6.5-030		
ASC	C++	Intel			8.1	
ASC	CAVE library	VRCO				
ASC	Cerius2	Accelrys	4.10			
ASC	CFD ++	metacomp	5.1	5.1	5.1	5.1
ASC	CHARMm	Accelrys	31b1			
ASC	Chimera Tools	NASA Advanced Supercomputing Division	1.9	1.9	1.9	
ASC	Cmake	Kitware, Inc.	2.4.2		2.4.2	2.4.2
ASC	Cobalt	Cobalt Solutions	3.0	3.0	3.0	
ASC	Cproto	Chin Huang/Thomas Dickey	4.6	4.6	4.6	4.6
ASC	CRYSTAL	Universita degli Studi di Torino	03	03	03	03
ASC	CTH	Sandia National Laboratory	Interim 03	Interim 03	Interim 03	
ASC	CVS	Concurrent Versions System	1.12.13	1.12.13	1.12.13	1.12.13
ASC	Dakota	Sandia	3.3			
ASC	DARWIN	Southwest Research Int'l.	5.1			
ASC	DDD	Free Software Foundation	3.3.11	3.3.11	3.3.11	
ASC	Deja Gnu	Free Software Foundation	1.4.2	1.4.2		1.4.2
ASC	DL_Poly	Daresbury Laboratory	3.02/2.14	3.02/2.14	3.02/2.14	
ASC	EADSIM	Teledyne Brown Engineering	12.0			
ASC	Emacs	Free Software Foundation	21.4.1	21.3.1	21.4.1	
ASC	Enscript	Free Software Foundation	1.6.4	1.6.4	1.6.4	
ASC	Ensight	CEI	8.0.7n		8.0.7n	8.0.7n
ASC	Expect	Don Libes/NIST	5.39.0	5.39.0	5.39.0	5.39.0
ASC	FAST	COSMIC				
ASC	FFTW	MIT	2.1.5	2.1.5	2.1.5	2.1.5
ASC	FIELDVIEW	Intelligent Light	11.0	11.0	11.0	
ASC	Flex	Free Software Foundation	2.5.31	2.5.31	2.5.31	
ASC	Fluent	Fluent Inc.	6.2.26	6.2.26	6.2.26	6.2.26
ASC	FMD	Jim Lupo/Ruth Pachter		1.12.0		
ASC	Fontconfig	Keith Packard	2.2.94			
ASC	FORTRAN 77	SGI	7.4.3			
ASC	FORTRAN 77	Compaq		5.5a-3548		
ASC	FORTRAN 90	SGI	7.4.3			
ASC	FORTRAN 90	Compaq		5.5a-3548		
ASC	FreeType	FreeType Project	2.1.4	2.1.4		
ASC	GAMESS	Iowa State University	27Jun05	27Jun05	27Jun05	22 Feb 06
ASC	GASP	Aerosoft	4.2.2	4.2.2	4.2.2	
ASC	Gaussian	Gaussian	03 D01	03 D01	03 D01	
ASC	Gaussview	Gaussian, Inc.	3.10	3.09	3.09	
ASC	GCC	Free Software Foundation	3.4.0	3.4.0		
ASC	GDB	Free Software Foundation	6.4	6.4	6.4	
ASC	gettext	Free Software Foundation	0.14.1	0.14.1	0.14.5	
ASC	Ghostscript	Aladdin Enterprises	8.53	8.53	8.53	8.53
ASC	Ghostview	Free Software Foundation	1.5	1.5	1.5	1.5
ASC	GIMP	Free Software Foundation				
ASC	GLIB	Free Software Foundation	2.6.3			
ASC	glibmm	Free Software Foundation	2.4.5	2.4.5		
ASC	gmp	Free Software Foundation	4.2.1	4.2.1	4.2.1	
ASC	GNU diffutils	Free Software Foundation	2.8.1	2.8.1	2.8.1	
ASC	GNU findutils	Free Software Foundation	4.2.27	4.2.27	4.2.27	4.2.27
ASC	GNU m4	Free Software Foundation	1.4.4	1.4.4	1.4.4	
ASC	GNU make	Free Software Foundation	3.81	3.81	3.81	

Table VII: DREN Software Summary – ERDC

Site	Name	Vendor	SGI Origin 2000 (C)	SGI Onyx 340	SGI Origin 3000	Compaq SC45
ERDC	ABAQUS	HKS			6.5-4	6.5-4
ERDC	Access / Seacas	Sandia			01 Jul 2005	01 Jul 2005
ERDC	ACML	AMD				
ERDC	ALE 3D	LLNL				4.0.0
ERDC	ALEGRA	Sandia			4.0.1	4.1
ERDC	ANACAP-U	Anatech			2.5-9	
ERDC	AVS/Viz Express	Advanced Visual System		6.0		
ERDC	AVS/Viz Express Multi-Pipe Edition	Advanced Visual System		6.0		
ERDC	AVUS	AFRL/VAAC			01 Jan 2004	
ERDC	C compiler	Compaq/Cray/SGI	7.4	7.4.3m	7.4.4m	6.5-303
ERDC	C++ compiler	Compaq/Cray/SGI	7.4	7.4.3m	7.4.4m	7.1-006
ERDC	CAVE Libraries	VRCO		3.0.3		
ERDC	CFD++	Metacomp			5.1.1	5.1.1
ERDC	Chimera Grid Tools	NASA AMES			1.9	
ERDC	Cobalt	Cobalt			3.0 (com)	3.0 (com)
ERDC	CPMD	CPMD Consortium			3.9.1	3.9.1
ERDC	Craylibs	CRAY				sciport (in dxml)
ERDC	CrayPat	CRAY				
ERDC	CrayTools	CRAY				
ERDC	Crunch	Craft Tech				
ERDC	CTH (restricted to approved personnel)	Sandia			7.1	7.1
ERDC	CVS	GNU		1.11.5	1.11.5	1.12.9
ERDC	Dimemas	CEPBA UPC			2.3	2.3
ERDC	DXML	Compaq				current
ERDC	Dysmas	NSWC Indian Head			4.30.5a	
ERDC	Earth Vision	DG		7.1		
ERDC	EnLighten Gold	CEI		8.0.7m	8.0.7e	8.0.7e
ERDC	EnSight Gold	CEI		8.0.7m	8.0.7e	8.0.7e
ERDC	EnVideo	CEI		8.0.22	8.0.14	
ERDC	FAST	NASA AMES		1.3		
ERDC	FFTW	MIT			3.1	3.1
ERDC	FieldView	Intelligent Light		11		
ERDC	Fluent	Fluent			6.2.16	
ERDC	Fortran 77/90 compiler	Compaq/Cray/SGI	7.4	7.4.3m	7.4.4m	5.6
ERDC	FTA	Platform		1.1	1.1	1.1
ERDC	GAMESS	ISU			16 Feb 2002 (R4)	26 OCT 2000
ERDC	GASP	AeroSoft			4.2.1	4.2.1
ERDC	Gaussian03 / Linda	Gaussian Inc.			03	
ERDC	Gaussian98	Gaussian Inc.			98	98
ERDC	GCC	GNU		3.3	3.2.2	3.4.0
ERDC	GDB	GNU			5.2	
ERDC	Gnu Utilities	Free Software Foundation			current	current
ERDC	Griz	LLNL			15 Oct 2001	
ERDC	HDF4	NCSA			4.1r2	4.1r5
ERDC	HDF5	NCSA			5-1.6.5	5-1.6.5
ERDC	IDL	RSI		6.2		
ERDC	IMSL	Visual Numerics			5.0	5.0
ERDC	IRIX64	SGI	6.5.23f	6.5.28f	6.5.28f	
ERDC	Java	Sun Microsystems		1.4.1	1.4.1	1.4.2
ERDC	KAP Pro Assure/Guide	KAI			4.0	4.0
ERDC	kf77/kf90	Compaq/HP				5.6
ERDC	Ladebug	Compaq/HP				4.0.69
ERDC	LAPACK	University of TN			1.4.1.3	5.2
ERDC	Libsci	CRAY				
ERDC	Lightwave	NewTek		5.6		
ERDC	LS-Dyna	LSTC			970 rev 5434a	
ERDC	LSF	Platform			6.0 HPC	6.0 HPC
ERDC	Mass Storage Utilities	ERDC MSRC		1.30	1.30	1.30
ERDC	Matlab (Math and Statistics)	Math Works		6.5.0 R13		
ERDC	Message Passing Interface (MPI)	Compaq/Cray/SGI		1.9	1.9	1.7
ERDC	ModSAF	DARPA			5.0	
ERDC	MOLPRO	UofB			2000.1	2000.1
ERDC	Motif	Open Group		2.1		
ERDC	mpich	Argonne			1.2.2	
ERDC	MPIDtrace	CEPBA UPC			1.0	1.0
ERDC	MPSCP	Sandia			1.2	1.2
ERDC	NCARG	NCAR			4.1.1	4.2.2
ERDC	NetCDF	Unidata			3.6.0	3.6.0
ERDC	Nike3D	LLNL			2000	
ERDC	NWchem	PNNL			4.5	4.5
ERDC	Octopus	Octopus Consortium				
ERDC	OmniORB	GNU			4.0.2	4.0.1
ERDC	OpenMP			1.9	1.9	1.1.2.2

Table VIII: DREN Software Summary – MHPCC

Site	Name	Vendor	IBM P3/P4/1.3
MHPCC	Cobalt	Cobalt Solutions Inc.	3
MHPCC	Gamess	Iowa St. Univ.	9.03
MHPCC	Gaussian 03	Gaussian, Inc.	g03.b4
MHPCC	IDL	Research Systems	6
MHPCC	Matlab	Mathworks	6.51.199709
MHPCC	Parallel Tools	Numerical Algorithms Group	5.12
MHPCC	Totalview	Etnus, LLC	7.0.1
MHPCC	Totalview	Etnus, LLC	6.3.1

Table IX: DREN Software Summary – NAVO

Site	Name	Vendor	Linux	Sun	Compaq	SciVis Systems
NAVO	Acrobat	Adobe		4.0		4.0
NAVO	Arc/Info	ESRI	8.02			8.02
NAVO	ArcView GIS	ESRI	3.2			3.2
NAVO	ArcView Spatial	ESRI	1.1			1.1
NAVO	AVS	Advanced Visual Systems	5.4			5.4
NAVO	Cave Library	VRCO	2.7			2.7
NAVO	CTH	Sandia				
NAVO	CVT	Cray				
NAVO	ESSL	IBM				
NAVO	Fledermaus	Interactive Visual Systems	4.3.3a			4.3.3a
NAVO	FLEXlm	Macrovision	8.3b			8.3b
NAVO	ghostscript	Free Software Foundation				7.05
NAVO	ghostview	Free Software Foundation				1.5
NAVO	GMT	University of Hawaii	3.4.2			3.4.2
NAVO	GNU Plot	Free Software Foundation				
NAVO	HDF	NCSA				4.0
NAVO	HDF Lib	NCSA				
NAVO	HYDA	W.F. Baird				3.1
NAVO	IDL	Research Systems	5.2			5.2
NAVO	ImageMagick	ImageMagick	5.2.0			5.2.0
NAVO	IMAGINE	ERDAS		8.5		8.5
NAVO	IMSL IPT (F90/FNL)	Visual Numerics	4.0			
NAVO	Matlab	Mathworks				6.5.1
NAVO	Matlab IP Toolbox	Mathworks				4.1
NAVO	Matlab SP Toolbox	Mathworks				6.1
NAVO	MAYA	Alias|Wavefront	4.5			4.5
NAVO	MPICH	Argonne				
NAVO	MPT	Cray				
NAVO	MUSE	NIMA				2.1
NAVO	NCAR Graphics	NCAR	4.1.1			4.1.1
NAVO	NetCDF					
NAVO	NimaMuse	NIMA				
NAVO	Open Inventor	SGI				X
NAVO	PAPI	University of Tennessee Knoxville				
NAVO	Performer	OpenGL	3.0.1			3.0.1
NAVO	PERL					5.6.1
NAVO	PESSL	IBM				
NAVO	PhotoShop	Adobe				
NAVO	PMTOOLKIT	IBM				
NAVO	POE	IBM				
NAVO	Power Animator	Alias|Wavefront				9.0
NAVO	PSSP	IBM				
NAVO	PST			1.2.1.0		1.2.1.0
NAVO	PVWave	Visual Numerics	7.0			7.0
NAVO	TAU	University of Oregon				
NAVO	TotalView	Etnus				
NAVO	Vampir	Pallas				
NAVO	Vampirtrace	Pallas				
NAVO	Vizserver	OpenGL	1.0			1.0
NAVO	Wind	NPARC		X		

5.3. Acronyms and Abbreviations

Table X: Acronyms and Abbreviations

ACE/TAO	Adaptive Communications Environment / The ACE ORB
AFOSR	Air Force Office of Scientific Research
AFRL	Air Force Research Laboratory
AFRL/IFGB	AFRL, Information Directorate, Information Grid Cyber Operations Branch
AFRL/IFTC	AFRL, Information Directorate, Advanced Computing Technology Branch
API	Application Programmer Interface
ARL	Army Research Laboratory
ARMS	Adaptive and Reflective Middleware Systems
ARSC	Arctic Region Supercomputer Center
ASC	Aeronautical Systems Center
BAA	Broad Agency Announcement
BSCW	Basic Support for Cooperative Work
C2	Command and Control
CDRL	Contract Data Requirements List
COM	Component Object Model
CONOPS	Concept of Operations
CORBA	Common Object Request Broker Architecture
CPU	Central Processing Unit
CRAD	Contract Research and Development
DARPA	Defense Advanced Research Projects Agency
DDS	Distributed Database Services
DIS	Distributed Interactive Simulation,
DoD	Department of Defense
EDF	Earliest Deadline First
ESCHER	Embedded Systems Consortium for Hybrid and Embedded Research
FCS	Future Combat Systems

FLAMES	FLexible Analysis Modeling and Exercise System
GIG	Global Information Grid
HLA	High Level Architecture
HPCMP	High Performance Computing Modernization Program
ICE	Integrated Collaborative Environment
IP	Internet Protocol
JMETC	Joint Mission Environment Test Capability
MHPCC	Maui High Performance Computing Center
MICA	Mixed Initiative Control of Automa-teams
MoBIES	Model Based Integration of Embedded Systems
MUF	Maximum Urgency First
NAVO	Naval Oceanographic Office
NCO	Network Centric Operations
OEP	Open Experimental Platform
OEIP	Open Experiment Integration Platform
ONR	Office of Naval Research
OSD	Office of the Secretary of Defense
OTIF	Open Tool Integration Framework
PAT	Process Action Team
PCES	Program Composition for Embedded Systems
PRiSm	Product line Real-time Software component model
RMS	Rate Monotonic Scheduling
RRS	Rome Research Site
RT/CORBA	Real Time Common Object Request Broker Architecture
SEC	Software Enabled Control
SiSPI	Software intensive Systems Producibility Initiative
SOAP	Simple Object Access Protocol
SWTT	Software and Systems Test Track
TTA	Time-Triggered Architecture

UC-Berkeley	University of California Berkeley
XMI	XML Metadata Interchange
XML	Extensible Markup Language